CS380: Introduction to Computer Graphics

Lab Session 6

EUNJI HONG

Spring 2023
KAIST

Keyframe Animation

Assignment 5
Linear interpolation

Assignment 6
Catmull-Rom interpolation

Assignment 5: Keyframe Animation I

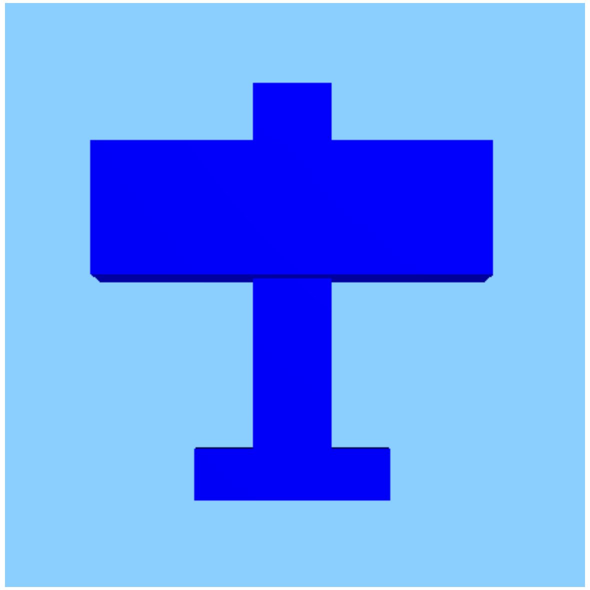
- In Assignment 5, the overall functions for managing a keyframe list and playing animation by interpolating the keyframes were implemented.
- The translation and rotation components of the RigTForms were linearly interpolated individually, using LERP for translation and SLERP for rotation.

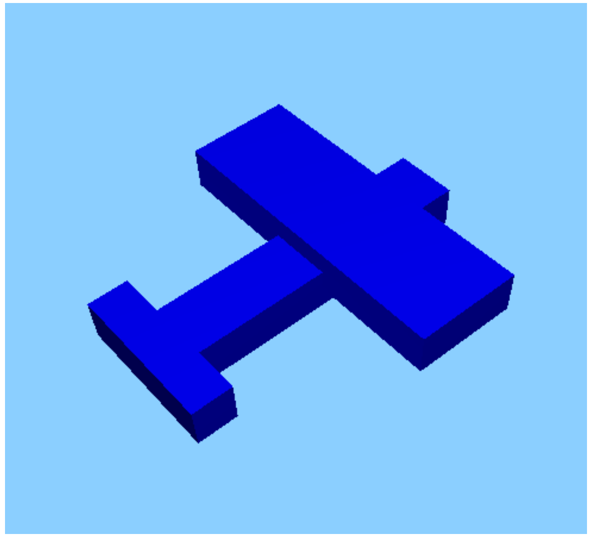
Assignment 6: Keyframe Animation II

- **Task**: Catmull-Rom interpolation
- Make an **airplane** which has a body and wings.
- Implement Catmull-Rom interpolation and play the animation.
- Perform an acrobatic flight including rotating and tilting.

Create an Airplane!

- In assignment 6, we will make and use an airplane.
- The airplane should include at least two boxes, one for the body and the other for the left and right wing.
- You can decorate it as you want, but it should still look like an airplane.
- **This is an additional specification not written in the PDF file.**



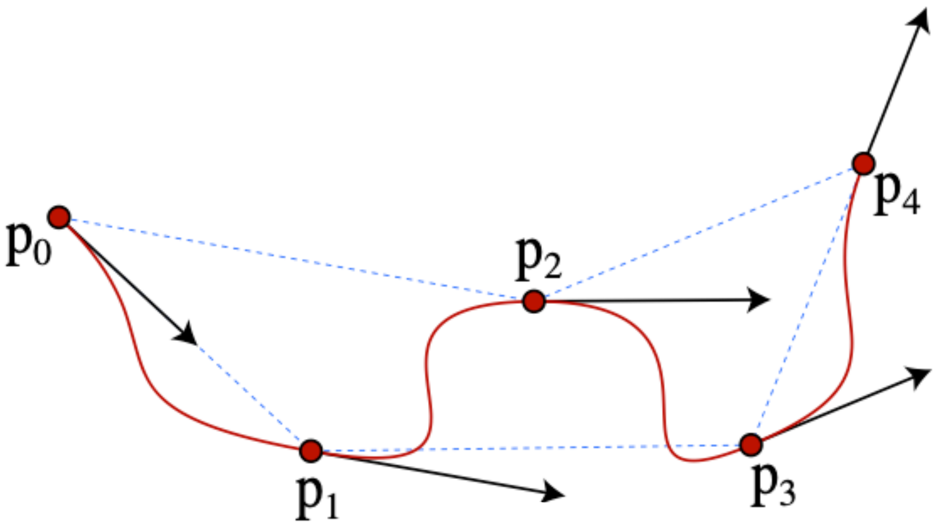


Setting up Assignment 6

- Assignment 6 will be built upon your assignment 5 codebase.
- Download the assignment 6 code file.
- Copy the new TODO function CatmullRom() from the downloaded interpolation.h file and paste it into your interpolation.h file.

Catmull-Rom Spline (CRS)

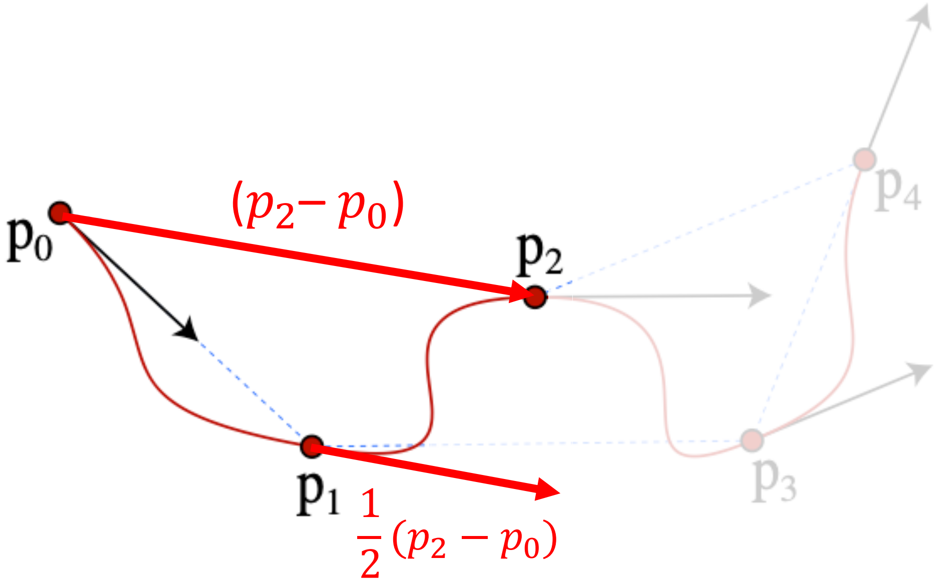
- It is an interpolating cubic spline with built-in C^1 continuity.
- It is formulated such that the tangent at each point p_i is calculated using the previous and the next point on the spline, $\frac{1}{2}(p_{i+1} - p_{i-1})$.



https://www.cs.cmu.edu/~fp/courses/graphics/asst5/catmullRom.pdf

Catmull-Rom Spline (CRS)

For example, the tangent at p_1 is defined as $\frac{1}{2}(p_2 - p_0)$.

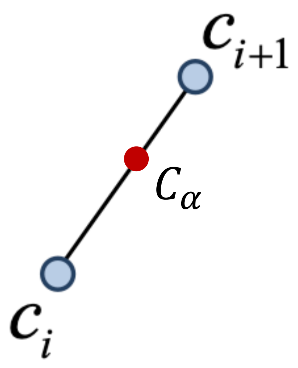


https://www.cs.cmu.edu/~fp/courses/graphics/asst5/catmullRom.pdf

CRS Interpolation

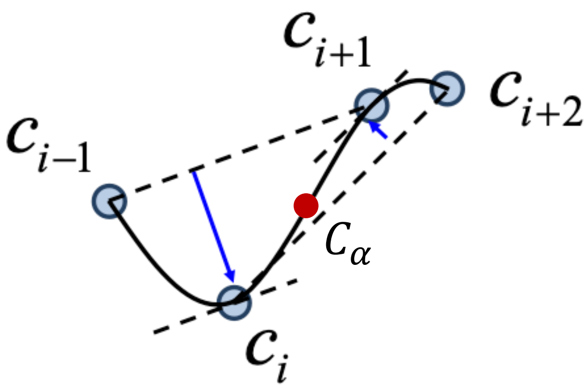
In assignment 6, we will interpolate the RigTForms in keyframes using Catmull-Rom interpolation.

Assignment 5



Linear interpolation

Assignment 6

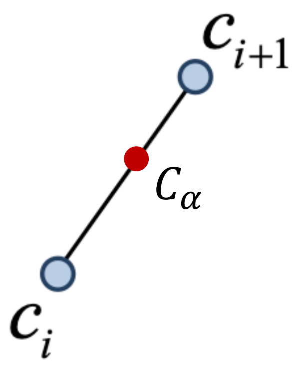


Catmull-Rom interpolation

CRS Interpolation

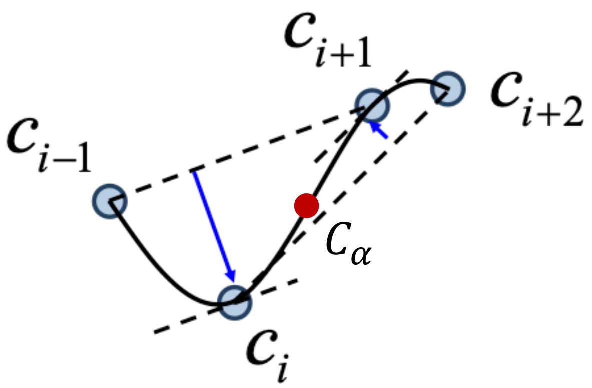
Similar to assignment 5, the translation and rotation components of RigTForm will be interpolated separately.

Assignment 5



Linear interpolation

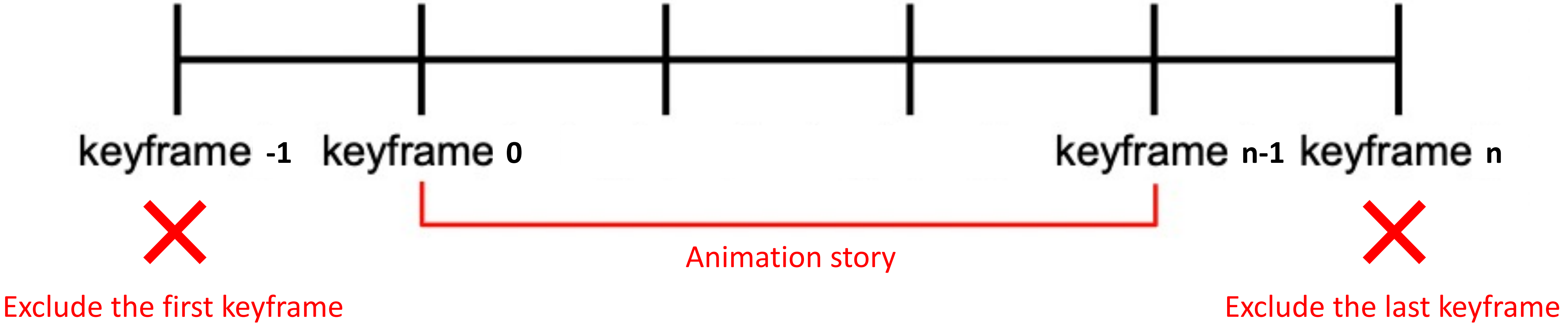
Assignment 6



Catmull-Rom interpolation

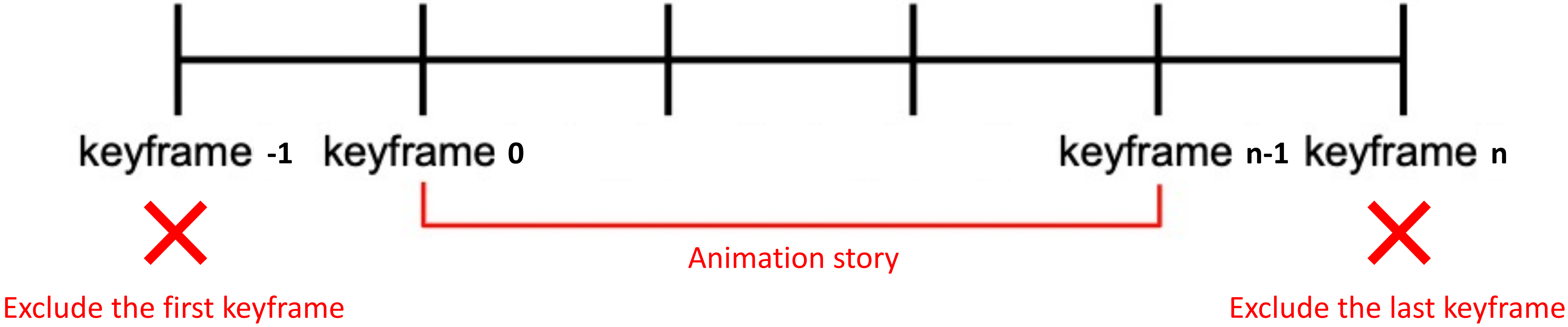
CRS Interpolation

Like Assignment 5, the animation is played by interpolating the keyframes excluding the first and the last keyframes.



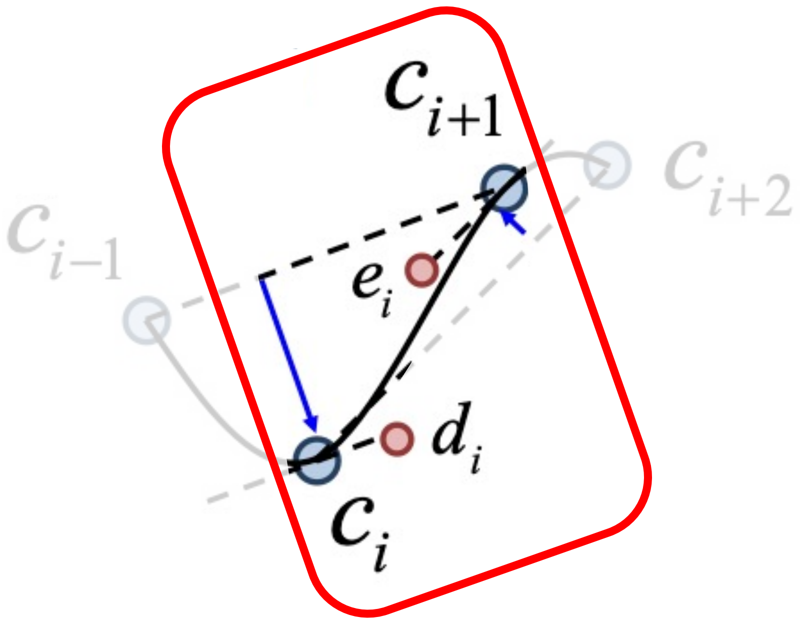
CRS Interpolation

So all the animation keyframes $\{\text{keyframe}_i\}$ $(i \in [0, n-1])$ can have their four keyframe set $(c_{i-1}, c_i, c_{i+1}, c_{i+2})$ that is required for Catmull-Rom interpolation.



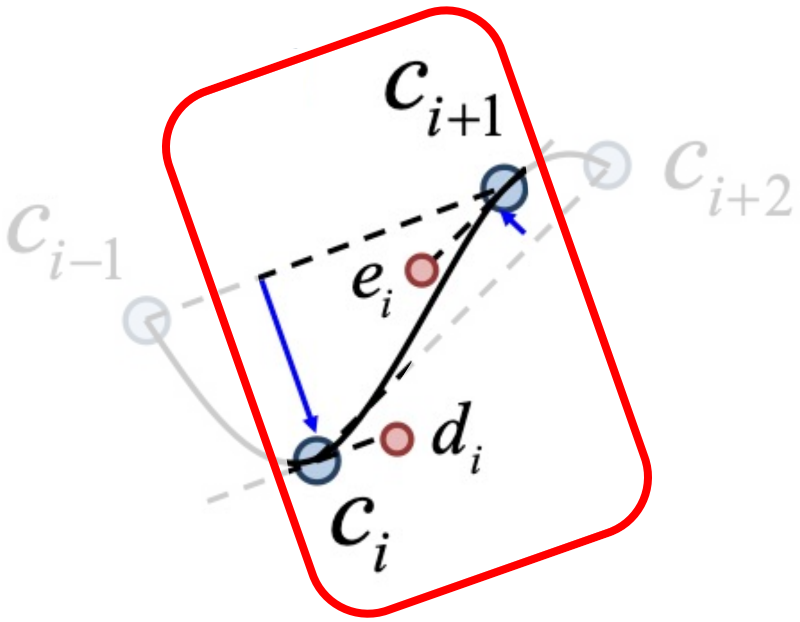
CRS Translation Interpolation

An interpolation between c_i and c_{i+1} assumes that two additional points d_i and e_i serve as cubic Bezier control points.



CRS Translation Interpolation

d_i and e_i are temporary points only for the interpolation and computed using the four control points c_{i-1}, c_i, c_{i+1}, and c_{i+2}.

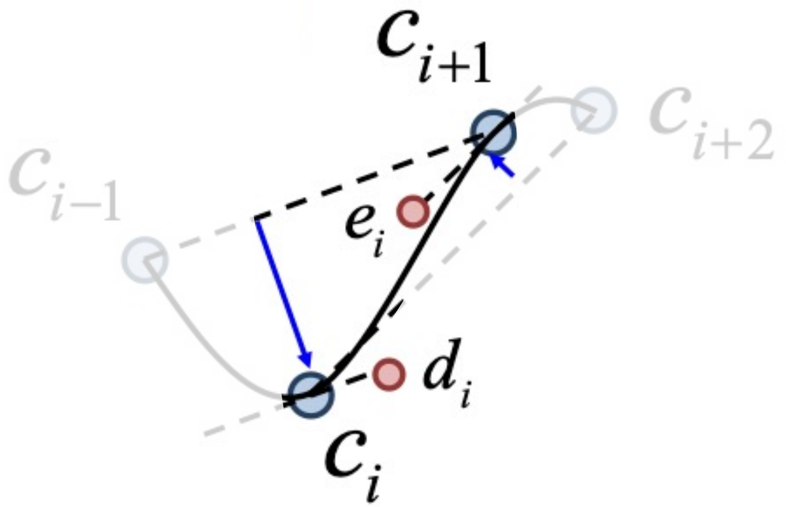


Let's see how we can represent d_i and e_i with the four control points c_{i-1}, c_i, c_{i+1}, and c_{i+2}!

CRS Translation Interpolation

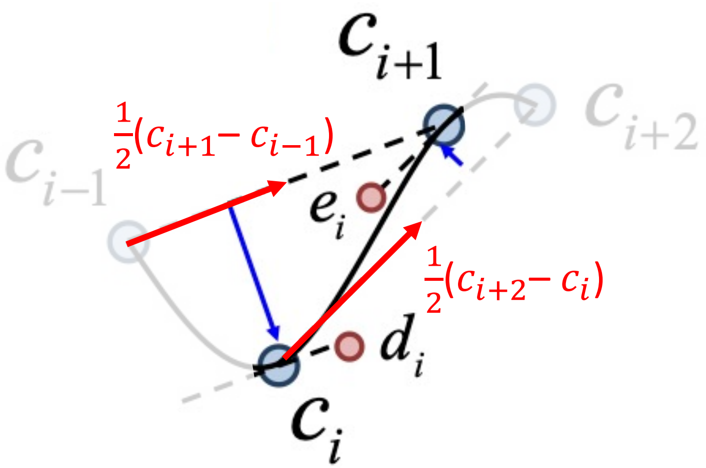
With four control points c_i, d_i, e_i, c_{i+1}, the cubic Bezier curve can be formulated as follows.

$$c(\alpha) = c_i(1-\alpha)^3 + 3d_i\alpha(1-\alpha)^2 + 3e_i\alpha^2(1-\alpha) + c_{i+1}\alpha^3$$



CRS Translation Interpolation

Catmull-Rom tangent condition of the curve:



$$c_i' = \frac{1}{2}(c_{i+1} - c_{i-1})$$

$$c_{i+1}' = \frac{1}{2}(c_{i+2} - c_i)$$

CRS Translation Interpolation

The first-order derivatives of the cubic Bezier curve:

$$c'(\alpha) = 3c_{i+1}\alpha^2 - 3e_i\alpha^2 - 3c_i(\alpha-1)^2 + 3d_i(\alpha-1)^2 - 6e_i\alpha(\alpha-1) + 3d_i\alpha(2\alpha-2)$$

$$\boxed{\begin{aligned} c_i' &= c'(0) = 3(d_i - c_i) \\ c_{i+1}' &= c'(1) = 3(c_{i+1} - e_i) \end{aligned}}$$

CRS Translation Interpolation

- Catmull-Rom tangent condition

$$c_i' = \frac{1}{2}\left(c_{i+1} - c_{i-1}\right)$$

$$c_{i+1}' = \frac{1}{2}\left(c_{i+2} - c_i\right)$$

- Cubic Bezier tangent condition

$$c_i' = 3\left(d_i - c_i\right)$$

$$c_{i+1}' = 3\left(c_{i+1} - e_i\right)$$

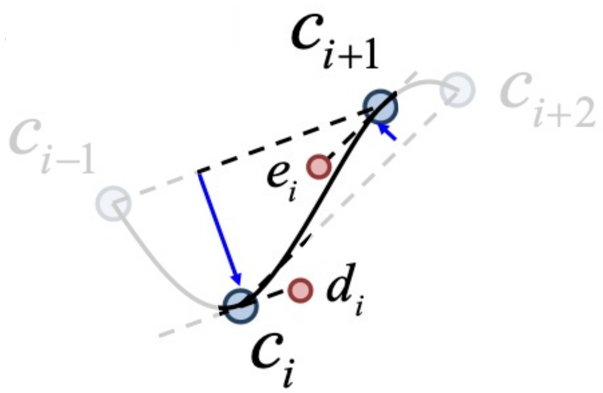
$$d_i = \frac{1}{6}\left(c_{i+1} - c_{i-1}\right) + c_i$$

$$e_i = -\frac{1}{6}\left(c_{i+2} - c_i\right) + c_{i+1}$$

CRS Translation Interpolation

Cubic Bezier curve with four control points c_i , d_i, e_i, c_{i+1}.

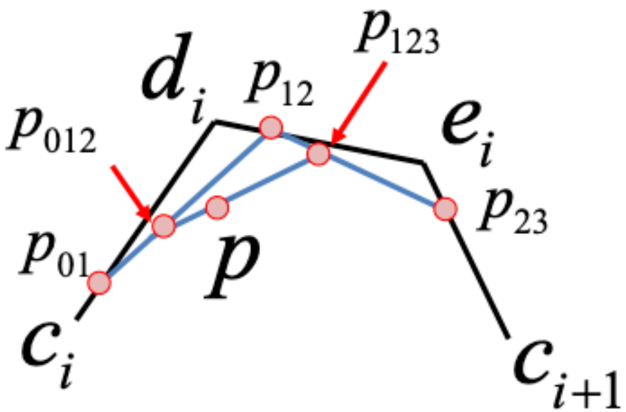
$$c(\alpha) = c_i(1-\alpha)^3 + 3d_i\alpha(1-\alpha)^2 + 3e_i\alpha^2(1-\alpha) + c_{i+1}\alpha^3$$



$$d_i = \frac{1}{6}\left(c_{i+1} - c_{i-1}\right) + c_i \qquad e_i = -\frac{1}{6}\left(c_{i+2} - c_i\right) + c_{i+1}$$

CRS Translation Interpolation

With the lerp(p_0, p_1, α) function from assignment 5, the interpolation point p can be easily calculated with the following chain of computations.



$$p_{01} = \text{lerp}(c_i, d_i, \alpha)$$

$$p_{12} = \text{lerp}(d_i, e_i, \alpha)$$

$$p_{23} = \text{lerp}(e_i, c_{i+1}, \alpha)$$

$$p_{012} = \text{lerp}(p_{01}, p_{12}, \alpha)$$

$$p_{123} = \text{lerp}(p_{12}, p_{23}, \alpha)$$

$$\boxed{p = \text{lerp}(p_{012}, p_{123}, \alpha)}$$

CRS Rotation Interpolation

- Scalar addition ⇔ Quaternion multiplication
- Scalar negation ⇔ Quaternion inversion
- Scalar multiplication ⇔ Quaternion power

$$d_i = \frac{1}{6}\left(c_{i+1} - c_{i-1}\right) + c_i$$

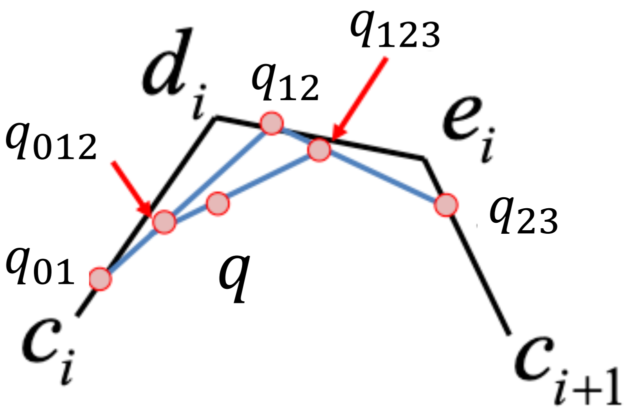
$$e_i = -\frac{1}{6}\left(c_{i+2} - c_i\right) + c_{i+1}$$

$$d_i = \left(\left(c_{i+1} c_{i-i}^{-1}\right)^{1/6}\right) c_i$$

$$e_i = \left(\left(c_{i+2} c_i^{-1}\right)^{-1/6}\right) c_{i+1}$$

CRS Rotation Interpolation

With the slerp(q_0, q_1, α) function from assignment 5, interpolation point q can be easily calculated with the following chain of computations.



$$q_{01} = \text{slerp}(c_i, d_i, \alpha)$$

$$q_{12} = \text{slerp}(d_i, e_i, \alpha)$$

$$q_{23} = \text{slerp}(e_i, c_{i+1}, \alpha)$$

$$q_{012} = \text{slerp}(q_{01}, q_{12}, \alpha)$$

$$q_{123} = \text{slerp}(q_{12}, q_{23}, \alpha)$$

$$\boxed{q = \text{slerp}(q_{012}, q_{123}, \alpha)}$$

Animation by CRS Interpolation

- The frame interpolation function in assignment 5 should be replaced with Catmull-Rom interpolation.
- The animation should continue to smoothly play after the replacement.

Animation Recordings

- Make a keyframe sequence with at least 8 keyframes.
- Record **two videos** of the keyframe animations:
  - one using the linear interpolation in assignment 5 and
  - the other using the Catmull-Rom interpolation in assignment 6.
- The videos should show the distinguishable difference in the animations.
- Create the two videos in .mp4 or .mov files.

Evaluation

- Is the Catmull-Rom function implementation correct?
- Is the animation working without any problem?
- Are the videos showing the distinguishable difference between the linear interpolation and the Catmull-Rom interpolation?

Submission

- Due: Sun, May 21 23:59 KST.
- Late submission: Up to two days (~Tues, May 23 23:59 KST) with a penalty of 20% of the score.
- Compress the codes and the videos in a .zip file and submit it on the GradeScope.